

P
526

Advance 86

Hardware Overview

Contents

Chapter 1	Advance 86 ROM BIOS	Page 1-1
1.1	Software Interrupts	1-1
1.2	Keyboard Encoding	1-3
1.3	Shift States	1-6
1.4	Special Handling	1-7
1.5	BIOS Calls	1-8
1.6	Advance 86 Memory Map	1-21

Index

CHAPTER ONE Advance 86 ROM BIOS

1.0 The Advance 86 ROM Basic Input/Output System resides in the top 8K of ROM on the Advance 86 system board. BIOS routines allow assembly language programs to control the hardware of the Advance 86 and perform I/O operations via the keyboard, video, cassette, printer, serial and disk subsystems.

The BIOS provides a degree of insulation from the hardware for the programmer. New peripherals can be added to the system, or production changes made without affecting compatability with your programs. If you do not go through the BIOS interface but instead choose to control the hardware directly you should be aware that there is no guarantee that your software will work on future versions of the Advance.

1.1 **Software Interrupts** BIOS routines are called through 8086 software interrupts. Each BIOS routine has its own interrupt vector. Table 1 shows the software interrupt assignments on the Advance 86.

Table 1. Advance 86 BIOS Software Interrupts

Interrupt Number	Address (Hex)	Purpose
00	00-03	Divide by 0
01	04-07	Single step
02	08-0B	NMI
03	0C-0F	Breakpoint (debug uses)
04	10-13	Overflow
05	14-17	Screen print
06	18-1B	Reserved
07	10-1F	Reserved
08	20-23	Timer Interrupt
09	24-27	Keyboard Interrupt
0A	28-2B	Reserved
0B	2C-2F	Communications Interrupt
0C	30-33	Communications Interrupt
0D	34-37	Reserved
0E	38-3B	Disk Interrupt
0F	3C-3F	Reserved Printer
10	40-43	Video I/O
11	44-47	Equipment attached
12	48-4B	Memory size
13	4C-4F	Disk I/O
14	50-53	Serial I/O
15	54-57	Cassette I/O
16	58-5B	Keyboard I/O
17	5C-5F	Printer I/O
18	60-63	Enter cassette Basic
19	64-67	Bootstrap loader
1A	68-6B	Timer I/O
1B	6C-6F	CTRL — BREAK
1C	70-73	Timer Tick
1D	74-77	Video init table
1E	78-7B	Disk base parameter table
1F	7C-7F	Graphics characters pointer

Note that the other vectors documented in Table 1 are central to the correct operation of the machine.

The BIOS I/O routines are given in section 1.5, BIOS calls.

Parameters are passed and returned in the 8086 registers.

1.2 Keyboard Encoding

F1 59	F2 60	ESC	1	2	3	4	5	6	7	8	9	0	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480
-------	-------	-----	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

The keyboard routine in ROM BIOS converts keyboard scan codes into 'extended' ASCII.

Table 2 shows the character codes which are passed through the BIOS keyboard routine to the application program.

Note that 'NA' indicates that the combination is suppressed in the keyboard routine. The two byte extended codes consist of ASCII 0, plus a second code as indicated.

Table 2. Character Codes

Key	LOWER CASE	UPPER CASE	CTRL	ALT
1	ESC	ESC	ESC	NA
2	1	1/2	NA	(0,120)
3	2	@	(0,3)	(0,121)
4	3	#	NA	(0,122)
5	4	\$	NA	(0,123)
6	5	%	NA	(0,124)
7	6	^	(030)	(0,125)
8	7	&	NA	(0,126)
9	8	*	NA	(0,127)
10	9	(NA	(0,128)
11	0)	NA	(0,129)
12	-	_	(031)	(0,130)
13	=	+	NA	(0,131)
14	(008)	(008)	(127)	NA
15	(009)	(0,15)	NA	NA
16	q	Q	(017)	(0,16)
17	w	W	(023)	(0,17)
18	e	E	(005)	(0,18)
19	r	R	(018)	(0,19)
20	t	T	(020)	(0,20)
21	y	Y	(025)	(0,21)
22	u	U	(021)	(0,22)
23	i	I	(009)	(0,23)
24	o	O	(015)	(0,24)
25	p	P	(016)	(0,25)
26	[{	(027)	NA
27]	}	(029)	NA
28	(013)	(013)	(010)	NA
29	NA	NA	NA	NA
30	a	A	(001)	(0,30)
31	s	S	(019)	(0,31)
32	d	D	(004)	(0,32)
33	f	F	(006)	(0,33)
34	g	G	(007)	(0,34)

532

Key	LOWERCASE	UPPERCASE	CTRL	ALT
35	h	H	(008)	(0,35)
36	j	J	(010)	(0,36)
37	k	K	(011)	(0,37)
38	l	L	(012)	(0,38)
39	.	:	NA	NA
40	'	"	NA	NA
41	`	~	NA	NA
42	NA	NA	NA	NA
43	\		(028)	NA
44	z	Z	(026)	(0,44)
45	x	X	(024)	(0,45)
46	c	C	(003)	(0,46)
47	v	V	(022)	(0,47)
48	b	B	(002)	(0,48)
49	n	N	(014)	(0,49)
50	m	M	(013)	(0,50)
51	,	<	NA	NA
52	/	>	NA	NA
53	/	?	NA	NA
54	NA	NA	NA	NA
55	★	PrtSc (See 1.4)	(0,114)	NA
56	NA	NA	NA	NA
57	SPACE	SPACE	SPACE	SPACE
58	NA	NA	NA	NA
59	(0,59)	(0,84)	(0,94)	(0,104)
60	(0,60)	(0,85)	(0,95)	(0,105)
61	(0,61)	(0,86)	(0,96)	(0,106)
62	(0,62)	(0,87)	(0,97)	(0,107)
63	(0,63)	(0,88)	(0,98)	(0,108)
64	(0,64)	(0,89)	(0,99)	(0,109)
65	(0,65)	(0,90)	(0,100)	(0,110)
66	(0,66)	(0,91)	(0,101)	(0,111)
67	(0,67)	(0,92)	(0,102)	(0,112)
68	(0,68)	(0,93)	(0,103)	(0,113)
69	NA	NA	Pause (See 1.4)	NA
70	NA	NA	Break (See 1.4)	NA

X

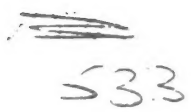


Table 3 gives the character codes for keys 71 to 83. Note that these keys only have meaning in lowercase, in Numlock (or shifted) or in Ctrl states. The shift key reverses temporarily the current Numlock states.

Table 3

Key	Numlock	Lowercase	ALT	CTRL
71	7	(0,71)	NA	(0,117)
72	8	(0,72)	NA	NA
73	9	(0,73)	NA	(0,119)
74	-	=	NA	NA
75	4	(0,75)	NA	(0,115)
76	5	NA	NA	NA
77	6	(0,77)	NA	(0,116)
78	+	+	NA	NA
79	1	(0,79)	NA	(0,117)
80	2	(0,80)	NA	NA
81	3	(0,81)	NA	(0,118)
82	0	Ins	NA	NA
83	-1	(0,83)	See 1.4	See 1.4

1.3 Shift States

The keyboard routine handles the shift states transparently to the application program. The current set of active shift states are available by calling INT 16H (keyboard I/O)

Table 4. Shift States

Key	Functions
Shift	Shifts keys 2-13, 15-27, 30-41, 43-53, 55, and 59-68 to uppercase. (Lowercase if in Caps-Lock state). Reverses the Num Lock or non-Num-Lock state of keys 71-73, 75, 77, and 79-83.
Ctrl	Shifts keys 3, 7, 12, 14, 16-28, 30-38, 43-50, 55, 59-71, 73, 75, 77, 79 and 81 to the Ctrl state. Used with the Scroll Lock key to cause 'break' function. Used with the Num Lock key to cause the 'pause' function.
Alt	Shifts keys 2-13, 16-25, 30-38, 44-50 and 59-68 to the Alt state. Alt allows you to enter any character code from 0 to 255 into the system from the keyboard. To do this, hold down the Alt key and type the decimal value of the characters desired using the numeric keypad. Note that you must type all three digits. Release the Alt key.

534

Caps Lock	Shifts keys 16-25, 30-38 and 44-50 to uppercase. A second press of key reverses the action.
Scroll Lock	INT 16H merely records the current shift state of the Scroll Lock key. Its interpretation is left to the applications program.

Note: If combinations of the Alt, Ctrl, and Shift keys are pressed, and only one is valid, the order of precedence is: the Alt key is first, the Ctrl key second and the Shift key third. The only valid combination is Alt and Ctrl which is used in the system reset.

1.4 Special Handling

System reset	Pressing the Alt, Ctrl and Del keys together will initiate a system reset.
Break	Pressing the Ctrl and Break keys together will initiate interrupt hex 1A. The extended characters (AL = hex 00, AH = 00) will be returned.
Pause	Pressing Ctrl and Num Lock keys together will cause the keyboard interrupt routine to loop and wait for any key (except Num Lock) to be pressed.
Print Screen	Pressing Shift and PrtSc (key 55) together will cause an interrupt to invoke the print screen routine.

535

1.5 BIOS CALLS

Print Screen (INT 5)

This interrupt prints a copy of the current text screen. It maintains screen status. It can be invoked from the DOS level by pressing Shift-PrtSc on the keyboard. A status byte at 0050:0000H is maintained by this routine.

0050:00000	= 0	Print screen has not been called or successful print screen operation.
	= 1	Print screen is in progress
	= 0FFH	Printer error

Video I/O (INT 10)

These routines provide the screen interface.

(AH)=0 Set screen mode
(AL)=0 40 x 25 BW
(AL)=1 40 x 25 Colour
(AL)=2 80 x 25 BW
(AL)=3 80 x 25 Colour
(AL)=4 320 x 200 Colour
(AL)=5 320 x 200 BW
(AL)=6 640 x 200 BW
(AL)=7 80 x 25 BW with external monochrome card.

The BW modes operate as colour modes, but the colour burst signal is not enabled.

(AH)=1 Set cursor type
(CH) = Bits 4-0 = start line for cursor
(CL) = Bits 4-0 = end line for cursor

(AH)=2 Set cursor
(DH,DL) = row, column
(BH) = page number

(AH)=3 Read cursor
(BH) = page number
On exit (DH,DL) = row, column of current cursor
(CH,CL) = cursor mode currently set.

(AH)=4 Read light pen
On exit
(AH) = 0 — light pen switch not down/not triggered
(AH) = 1 — valid light pen value in registers
(DH,DL) = row, column of character LP position
(CH) = Raster line (1-199)
(BX) = pixel column (0-319,639)

(AH)=5 Select current display page (Valid only for text modes)
(AL) = new page value (0-7 for 40 x 25, 0-3 for 80 x 25).

(AH)=6 Scroll current page up
(AL) = number of lines, input lines blanked at bottom of window

Note: AL = 0 means blank entire window
(CH,CL) = row, column of upper left corner of scroll
(DH,DL) = row, column of lower right corner of scroll
(BH) = character attribute to be used on blank line

38

In 40 x 25 or 80 x 25 text modes, the value set for palette colour 0 indicates the border colour to be used (0-31).

(AH)=12 Write pixel

(DX) = row number

(CX) = column number

(AL) = colour value

If bit 7 of AL = 1, then the colour value is exclusive OR'd with the current contents of the pixel.

(AH)=13 Read pixel

(DX) = row number

(CX) = column number

(AL) = returns the pixel read

(AH)=14 Write teletype

(AL) = character to write

(BL) = foreground colour in graphics mode

(BH) = display page in text mode

(AH)=15 Current video state

Returns the current video state

(AL) = mode currently set

(AH) = number of character columns on screen

(BH) = current display page

On INT 10 calls registers CS, SS, DS, ES, BX, CX, DX are preserved during call.

539

~~539~~

Equipment Determination (INT 11)

On exit

(AX) is set to indicate what peripheral devices are present on the system.

Bit 15, 14 = number of parallel ports

Bit 13 unused

Bit 12 = game I/O

Bit 11, 10, 9 = number of RS232 ports

Bit 8 unused

Bit 7, 6 = number of disk drives

00 = 1, 01 = 2.

Bit 5, 4 = video mode

00 — unused

01 — 40 x 25 BW using colour card

10 — 80 x 25 BW using colour card

11 — 80 x 25 BW using external monochrome card

Bit 3, 2 = Unused

Bit 1 = 50Hz/60Hz frame rate

Bit 0 = 86b expansion unit is attached

340
~~12~~

Memory Check (INT 12)

This routine calculates the amount of memory in the system.

One exit

(AX) = number of contiguous 1K blocks of memory.

541
un

Disk I/O (INT 13)

On entry

(AH)=0 Reset disk system

Hard reset to FDC.

(AH)=1 Read the disk status into (AL).

(DL) — Drive number (0-3)

(DH) — Head number (0-1)

(CH) — Track number (0-39)

(CL) — Sector number (1-9)

(AL) — Number of sectors (0-9)

(ES:BX) — Address of data buffer

(AH)=2 Read sectors into memory

(AH)=3 Write sectors from memory

(AH)=4 Verify sectors

(AH)=5 Format track.

For format, the buffer pointer (ES, BX) must point to the address fields for the track. Each field is composed of 4 bytes — track number, Head number, Sector number and Number of bytes per sector (00=128, 01=256, 02=512, 03=1024). There must be one entry for each sector on the track.

0:0078H — 0:007BH points to current disk parameters

On exit

AH = status

Status bits are defined in Table 5, page 1-26

Carry is set if operation failed.

Registers DS, BX, DX, CH, CL are preserved.

AL = number of sectors read

(AL may not be correct if time out error occurs).

Note: if an error is reported by the disk code, reset the disk, and retry the operation. On Read accesses, no motor start delay is allowed for so up to three retries are recommended on reads to ensure that the problem is not due to disk motor start-up.

RS232 I/O (INT 14)

(AH)=0 Initialise the serial port.
AL = initialisation parameters.

Bit	7	6	5	4	3	2	1	0
	----Baud rate---			- Parity--		Stopbit	Word Length	
	000 — 110			00 — None		0 — 1	10 — 7 Bits	
	001 — 150			01 — Odd		1 — 2	11 — 8 Bits	
	010 — 300							
	011 — 600							
	100 — 1200							
	101 — 2400							
	110 — 4800							
	111 — 9600							

On exit serial status is returned in (AX).

(AH)=1 Send the character in (AL)
(AL) register is preserved

On exit, bit 7 of AH is set if the routine was unable to transmit the character. The remainder of AH = the current status of the line.

(AH)=2 Receive a character in (AL)

On exit, AH has the current line status.

Error bits. If bit 7 = 1, data set ready was not flagged.

(AH)=3 Return the serial status in (AX).

AH contains the line status

Bit 7 = time out

Bit 6 = TSR empty

Bit 5 = THR empty

Bit 4 = break

Bit 3 = framing error

Bit 2 = parity error

Bit 1 = overrun error

Bit 0 = data ready

AL contains the modem control status

Bit 7 = received line signal detect

Bit 6 = ring indicator

Bit 5 = data set ready

Bit 4 = clear to send

Bit 3 = delta receive line signal detect

Bit 2 = trailing edge ring detector

Bit 1 = data set ready

Bit 0 = clear to send

On entry (DX) = Which RS232 port (0,1)

On exit AX Status, as above. All other registers are preserved.

Cassette I/O (INT 15)

(AH)=0 Turn cassette motor on

(AH)=1 Turn cassette motor off

(AH)=2 Read 1 256 byte blocks from cassette

(ES, BX) = pointer to data buffer

(CX) = count of bytes to read

On exit

(ES, BX) = pointer to last byte read + 1

(DX) = count of bytes read

Carry bit is set if an error occurred and nature of error.

(AH) = 01 if CRC error.

= 02 if data transitions lost.

= 04 if no data was found.

(AH)=3 Write 256 byte blocks to cassette

(ES, BX) = pointer to data buffer

(CX) = count of bytes to write

On exit

(EX, BX) = pointer to last byte written + 1

(CX) = 0

Keyboard I/O (INT 16)

On entry

- (AH)=0 Read next ASCII character. Return the ASCII code in (AL), scan code in (AH).
- (AH)=1 See if a keyboard character is available.
 - (ZF) = 0 — character is available
 - (ZF) = 1 — no character is available
 - (AX) = next available character bit remains in keyboard buffer
- (AH)=2 Return the current shift status in AL.
See Table 6 page 1-26

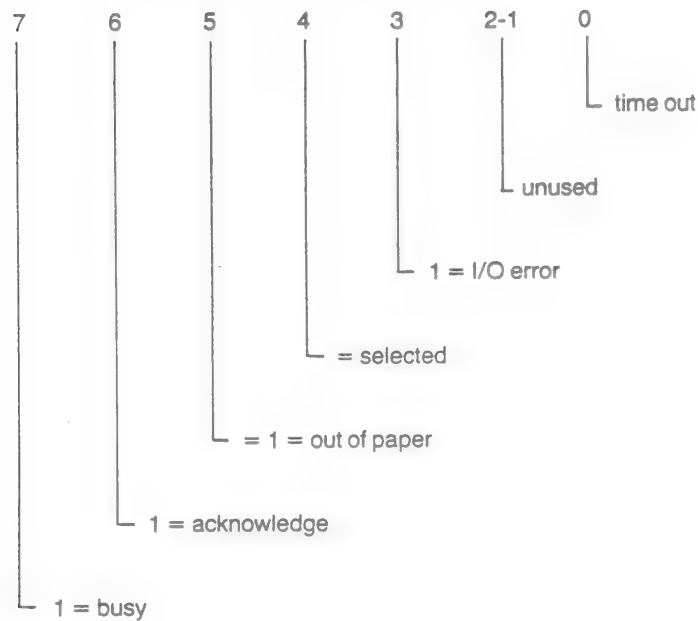
On exit

AX, Flag register changed. All other registers preserved.

Printer I/O (INT 17)

On entry

- (AH)=0 Send the character in (AL) to the parallel port.
On return, AH = 1 if character could not be sent. Other bits set as status call
- (AH)=1 Initialise the parallel port
Returns with (AH) set with status
- (AH)=2 Read the parallel port status into (AH)



(DX) = parallel port number (0-2).

On exit

AH = status. All other registers preserved.

Dugran
As on
draft

Bootstrap leader (INT 19)

On Advance 86b systems track 0, sector 1 is read into the boot location (0000:7C00) and control transferred there.

On an Advance 86a or if the disk boot fails, control is transferred to the cassette basic entry point.

Time of Day I/O (INT 1A)

On entry

(AH)=0 Return current clock setting

On exit CX = high portion of count

DX = low portion of count

AL = 0 if timer has not passed 24 hours since last read.

<> 0 if on another day.

(AH)=1 Set clock.

CX = high portion of count

DX = low portion of count

Note: Counts occur approximately 18.2 / sec.

548
cm

1.6 Advance 86 Memory Map

FFFF:0000	-----	Advance 86 Self-test routines, ROM BIOS (8K bytes)
FE00:0000	-----	Advance 86 Cassette Basic (56K bytes)
F000:0000	-----	Reserved
BC00:0000	-----	Video RAM (16K bytes)
B800:0000	-----	Reserved
A000:0000	-----	Top of possible RAM memory (640K bytes)
		Transient portion of COMMAND.COM (overlaid if necessary)
		Applications program
		Program segment prefix (100H bytes)
		Resident portion of COMMAND.COM
		Resident device drivers
		DOS Buffers
		MSDOS.SYS
		IO.SYS
0070:0000		ROM BIOS data
0040:0000		8086 Software Interrupt Vectors
0000:0000		

Advance 86 I/O Address Map**Address (Hex)**

00-0F	8237 DMA Chip
20	8259A Interrupt Controller
22★	8259A IMR Register; ★ or 21 on 86 Rev. 8 (redundantly decoded)
40 - 43	8253 TIMER
60 - 63	8255 PPI
80 - 83	DMA Page registers
AX	NMI mask register. 00 disable NMI. 80H enable NMI. Disabled at power on.
200 - 20F	Game I/O logic
378 - 37F	Parallel Printer Interface
3D0 - 3DF	6845 Video Logic
3F0 - 3F7	8272 Disk Controller Logic
3F8 - 3FF	8250 Asynchronous Communications Element

X = don't care

8237 DMA Controller

Channel Number	Purpose
0	I/O Channel RAM Refresh
1	Available on I/O Channel
2	Diskette Controller
3	Available on I/O Channel

8259A Programmable Interrupt Controller

Interrupt Level	Vector Number	Purpose
0	8	Timer Channel 0
1	9	Keyboard
2	A	Unused
3	B	Asynchronous Comms.
4	C	Reserved
5	D	Reserved
6	E	Diskette Controller
7	F	Parallel Printer

8253 Interval Timer

Timer Number	Purpose
0	Time of day interrupt
1	RAM refresh on I/O channel
2	Tone output

SSD
*rom***8255 PPI****Address (Hex)**

60 Input Keyboard scan codes
61 Output

- Bit 0 Timer channel 2 gate (speaker control)
1 Speaker (direct control)
2 Not used
3 Cassette motor off (active high)
4 Enable parity check on system memory (active low)
5 Enable parity check on I/O memory (active low)
6 Hold keyboard clock low (active low)
7 If high — clear keyboard; if low, enable keyboard

62 Input

- Bit 0 Switch 1 — B expansion unit fitted (1)
1 Switch 2 — 50hz (0), 60hz (1) frame rate
2 Switch 3 — Reserved
3 Switch 4 — Reserved
4 Cassette data in (active high)
5 Timer channel 2 (active high)
6 Parity fault on I/O memory (active high)
7 Parity fault on System memory (active high)

63 8255 Command Register initialised to 99H by ROM BIOS

Game I/O Logic

201H Input

- Bit 0 Analogue input 1
1 Analogue input 2
2 Analogue input 3
3 Analogue input 4
4 Button 1 (active low)
5 Button 2 (active low)
6 Button 3 (active low)
7 Button 4 (active low)

201H Output Begin conversion

To read game I/O, output any value to 0201H. Bits 0-3 will go low. After a time period proportional to the resistance value across it each bit will return to 1.

Parallel Printer Interface**Address**

378H Output

- Bit 0 DB0 Pin 2
1 DB1 Pin 3
2 DB2 Pin 4
3 DB3 Pin 5
4 DB4 Pin 6
5 DB5 Pin 7
6 DB6 Pin 8
7 DB7 Pin 9

379H Input

- Bit 0 Not used
- 1 Not used
- 2 Not used
- 3 Printer error (active low) Pin 15
- 4 Select (active high) Pin 13
- 5 Out of paper (active high) Pin 12
- 6 Acknowledge (active low) Pin 10
- 7 Busy (active high) Pin 11

37AH Output

- Bit 0 Strobe (active low) Pin 1
- 1 Auto LF (active low) Pin 14
- 2 Printer init. (active low) Pin 16
- 3 Printer select (active low) Pin 17
- 4 Enable printer interrupt★ (active high)
- 5 Not used
- 6 Not used
- 7 Not used

★ If bit 4 is 1, an interrupt will be generated when Acknowledge (Pin 10) goes low.

Video Logic

Address

3D4H 6845 CRT Controller Address Registers

3D5H 6845 CRT Controller Data Registers

3D8H Output Mode Select

- Bit 0 40 x 24 (0) or 80 x 25 (1) display
- 1 Text (0) or Graphics (1) display
- 2 Colour (0) or Black and white (1) display
- 3 Disable (0) or enable (1) the video display
- 4 320 x 200 (0) or 640 x 200 (1) display
- 5 16 background colours (0) or flashing in text modes (1)
- 6 Not used
- 7 Not used

3D9H Output Colour Select

- Bit 0 Blue border/background colour select
- 1 Red border/background colour select
- 2 Green border/background colour select
- 3 High intensity border/background colour select
- 4 Select high intensity background colours in text modes (active high)
- 5 Select Colour palette 0 or 1 in 320 x 200 graphics
- 6 Not used
- 7 Not used

Table 5.

Disk Status Byte.

Code (hex)	Error
01	Illegal command
02	Bad address mark
03	Write protect error
04	Sector not found
08	DMA overrun
09	DMA operation attempted to cross 64K boundary
10	CRC Read error
20	FDC failure
40	Seek failed
80	No response from FDC

Table 6.

Keyboard Shift Status

Bit position (Hex)	Function
0	Right shift key pressed
1	Left shift key pressed
2	CTRL key pressed
3	Alt key pressed
4	Scroll lock mode
5	Num Lock mode
6	Caps lock mode
7	Insert mode
8	Not used
9	Not used
A	Not used
B	Hold mode
C	Scroll lock pressed
D	Num Lock pressed
E	Caps lock pressed
F	Insert key pressed.

3DAH Input Video Status
Bit 0 Display buffer may be accessed in 80 x 25 mode (active high)
1 Light Pen Trigger set (active high)
2 Light Pen Switch (active low)
3 Alpha video signal

3DBH Output Clear Light Pen latch

3DCH Output Preset Light Pen latch

Floppy disk logic

Address

3F2H Output
Bit 0 Drive Select low 00-Drive A 01-Drive B
1 Drive Select high 10-Drive C 11-Drive D
2 Reset (active low)
3 Enable DMA + Interrupt requests (active high)
4 Drive A Motor enable (active high)
5 Drive B Motor enable (active high)
6 Drive C Motor enable (active high)
7 Drive D Motor enable (active high)

3F4H 8272 FDC Status Register

3F5H 8272 FDC Data Register

Asynchronous Communications logic

03F8H 8250 TX/RX/Divisor latch LSB
03F9H 8250 Divisor latch/Interrupt enable register
03FAH 8250 Interrupt identification register
03FBH 8250 Line control register
03FCH 8250 Modem control register
03FDH 8250 Line status register
03FEH 8250 Modem status register

Index

Advance 86 I/O Address Map 1-22
Alt key 1-6, 1-7
Asynchronous communications logic 1-25
Bootstrap leader (INT 19) 1-19
Break 1-7
Caps Lock key 1-6, 1-7
Cassette I/O (INT 15) 1-16
Character Codes 1-4 to 1-6
Clock setting 1-20
Ctrl key 1-6, 1-7
Disk I/O (INT 13) 1-14
Equipment Determination (INT 11) 1-12
Extended ASCII 1-4
Floppy disk logic 1-25
Game I/O logic 1-23
Keyboard 1-3
Keyboard Encoding 1-3 to 1-6
Keyboard I/O (INT 16) 1-17
Light Pen 1-9, 1-25
Memory check (INT 12) 1-13
Memory Map 1-21
Motor Start 1-14, 1-16
NumLock 1-6
Palette 1-10
Parallel printer interface 1-23
Pause 1-7
Print Screen 1-7
Print Screen (INT 5) 1-8
Printer I/O (INT 17) 1-18
RS232 I/O (INT 14) 1-15
Scroll Lock key 1-6, 1-7
Shift key 1-6, 1-7
Shift States 1-6 to 1-7
Special Handling 1-7
Software Interrupts 1-1, 1-2
System Reset 1-7
Time of Day I/O (INT 1A) 1-20
Video I/O (INT 10) 1-9
Video logic 1-24

version 5.5 =

BIBLIOGRAPHY

- Advance 86 User's Guide. Advance Technology. London 1984.
- iAPX 86, 88 User's Manual. Intel. Santa Clara CA 1981.
- MCS-86 Assembly Language Reference Manual. Intel. Santa Clara CA 1982
- The 8086 Book. Russell Rector & George Alexing. Osborne/Mcgraw Hill 1980
- The PC-DOS Handbook. Richard King. Sybex 1984.
- The 8086/8088 Printer. Steven Morse. Hayden 1982

Premier

182.